Managing Peer-to-Peer Applications

Vana Kalogeraki
Hewlett-Packard Laboratories
Palo Alto, CA 94304

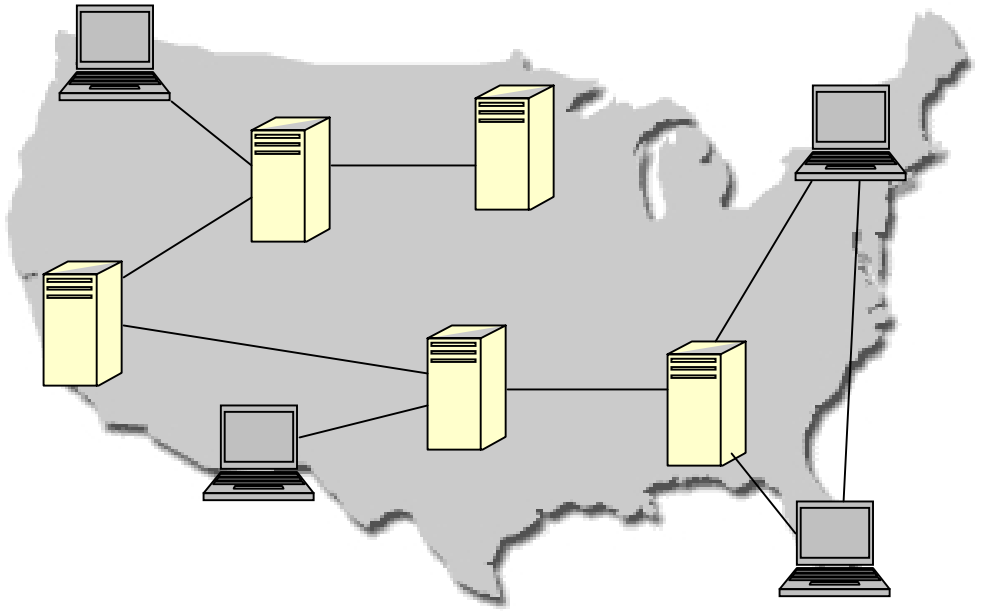
vana@hpl.hp.com

What is a P2P System?

"Sharing of computer resources and services through direct exchange"

- Virtual point to multipoint network of many peers
- Symmetric communications between peers
- Ad-hoc communication collaboration between peers

&



P2P Taxonomy

- Distributed Computations/Edge Services
  - Employ and aggregate unused processing power and storage resources of individual computers
  - Seti@Home, Intel, Entropia
- Distributed Search/Information Sharing
  - Message broadcasting system to discover peers and search for information
  - Gnutella, Freenet, Napster
- Distributed Computing Platform
  - Real-Time Collaboration
  - Sun's JXTA, Groove, Proksim

P2P Success Stories

- Popularity of Napster
  - Online users: 9,842
  - Shared MP3 files: 1,633,585
  - Shared drive space (GB): 6,838
- Popularity of Gnutella Network
  - Online peers: 43,546
  - Shared files: 1,843,549
  - Shared drive space (GB): 41,170
- Popularity of JXTA
  - Online users: 6,809
  - CVS Commits: 769

The Advantages are Compelling

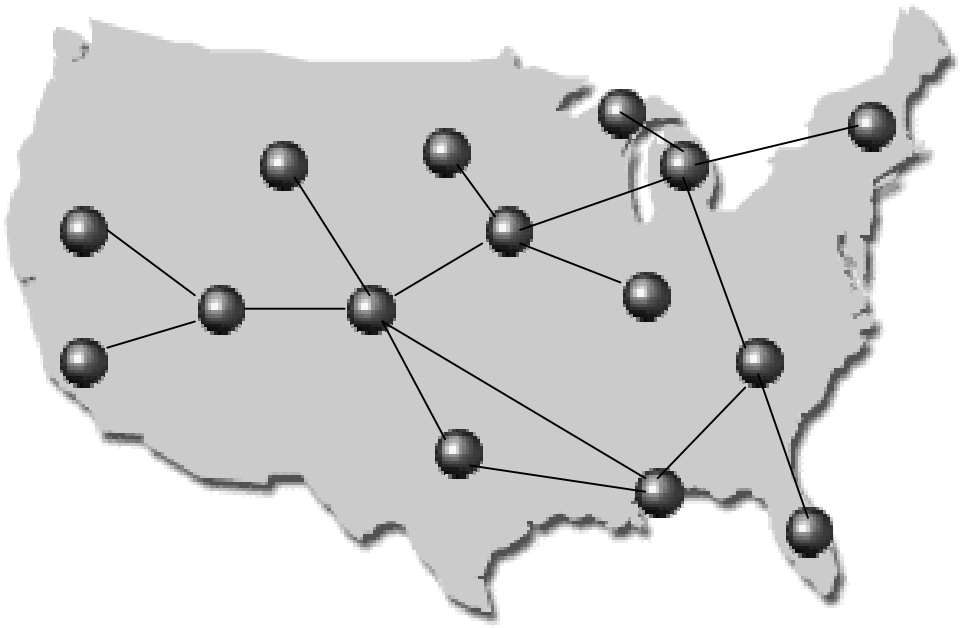
- Innovation at the edges of the network
- Cost savings
  - Aggregate existing resources
- Autonomous, decentralized systems
  - Direct & real-time connectivity
- High-availability
  - Remove single point of failure
- Various Transparencies
  - Location, migration, language/OS/platform

But what are the challenges?

- How to build a system that will scale with exponential network growth?
- How to efficiently propagate the requests through a large number of peers?
- How to monitor the actual behavior of the peers and the actual usage of the resources?
- How to dynamically deploy the objects on the peers and balance the load on the resources?
- How to provide end-to-end QoS and real-time guarantees?

Peer-to-Peer Applications

- Modeled as a sequence of method invocations of objects across multiple processors
- Topology of the resulting network could be random
- Dynamically setup routes
- Problem: which peers to connect to?

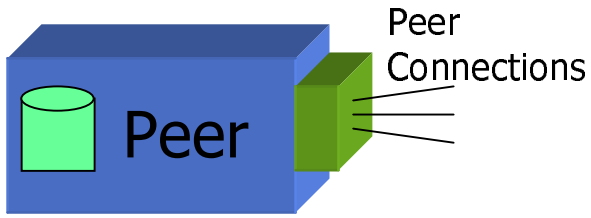


Issues in the P2P networks

- Architectural organization
- Routing of queries in the network
- These are difficult problems because..
  - Dealing with the dynamic aspects of the system
    - Peer arrival/departures
    - Data publishing/withdrawal
  - Peers have limited knowledge
  - Peers make autonomous decisions

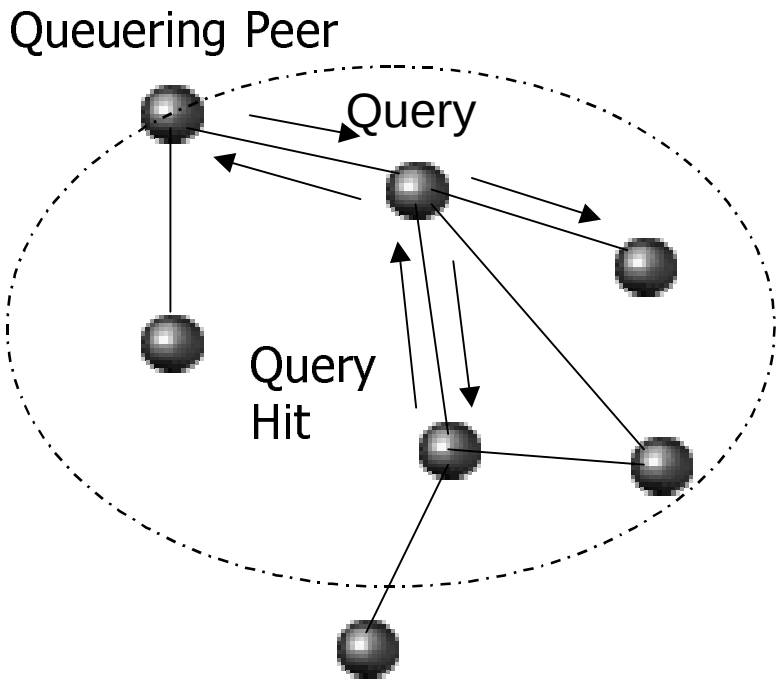
The Model

- Peer
  - Set of interests
  - List of peers
  - Horizon
  - Local Knowledge
  - Resource capabilities (communication, processing, storage)
  - Asynchronous requests for information
  - Timely delivery of critical content



Searching in the Network

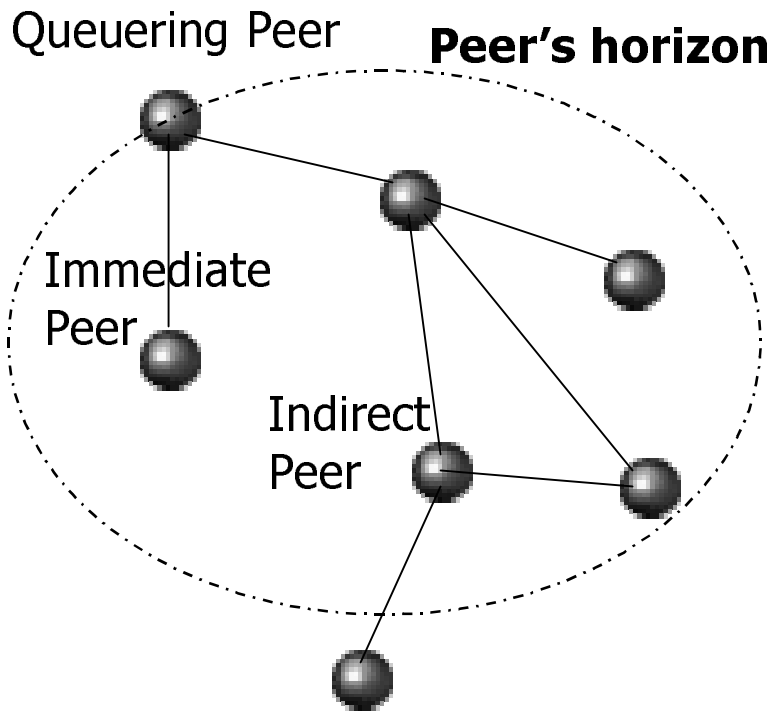
- Messages are sent over multiple hops from one peer to another
- P2P Protocol
  - Ping: discover active peers in the network
  - Pong: reply to a Ping message including IP address
  - Query: search for data in the network
  - QueryHit: reply to a Query message with the corresponding results



Peer Profiles

Monitor

- Search requests (type & time)
- Number & source of replies
- Time to get a result
- Immediate peer who propagated the results
- Peer failures



Resource Monitoring

- Current utilization of processor resources
  - processor (*i.e.,* cpu, memory)
  - storage (*i.e.,* disk)
- Bandwidth on the communication links
- Percentage of resources used by executing requests
- Profiling frequency is important
  - User behavior

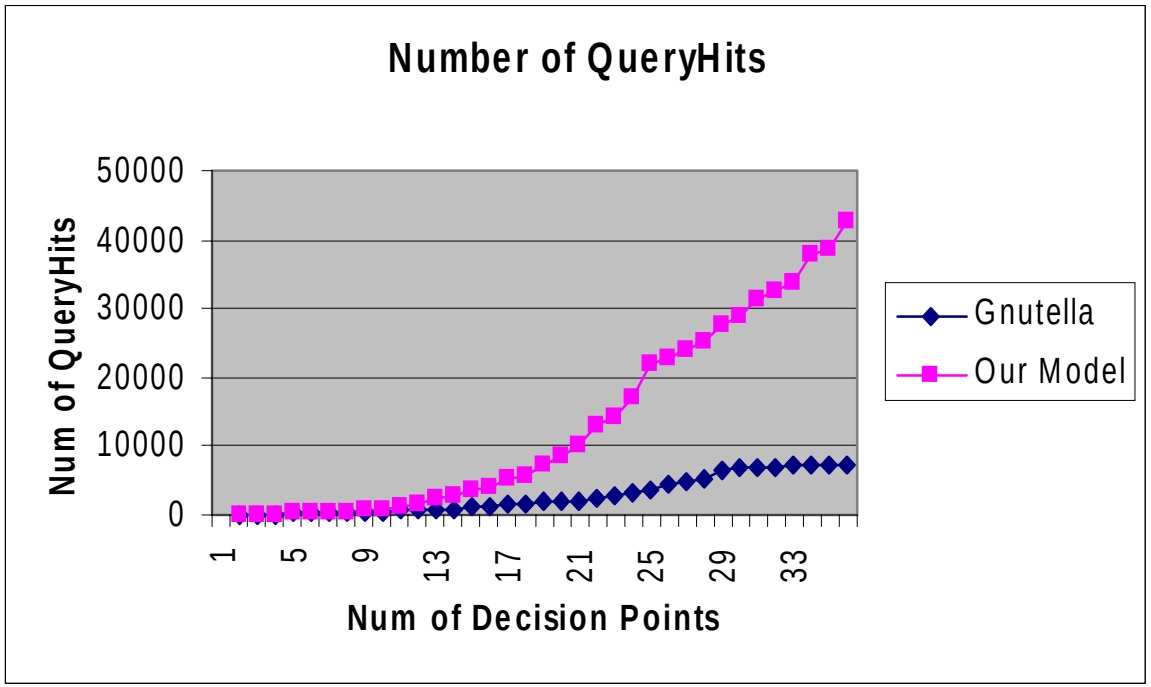
System Objectives

- Importance of a Peer

$$Importance_p(q,t) = a * \frac{percQueryHits_p(q)}{averNumHops_p(q)} + (1-a) * Importance_p(q,t-1)$$

- Minimize the number of messages in the network
- Retrieve data fast

Performance Results

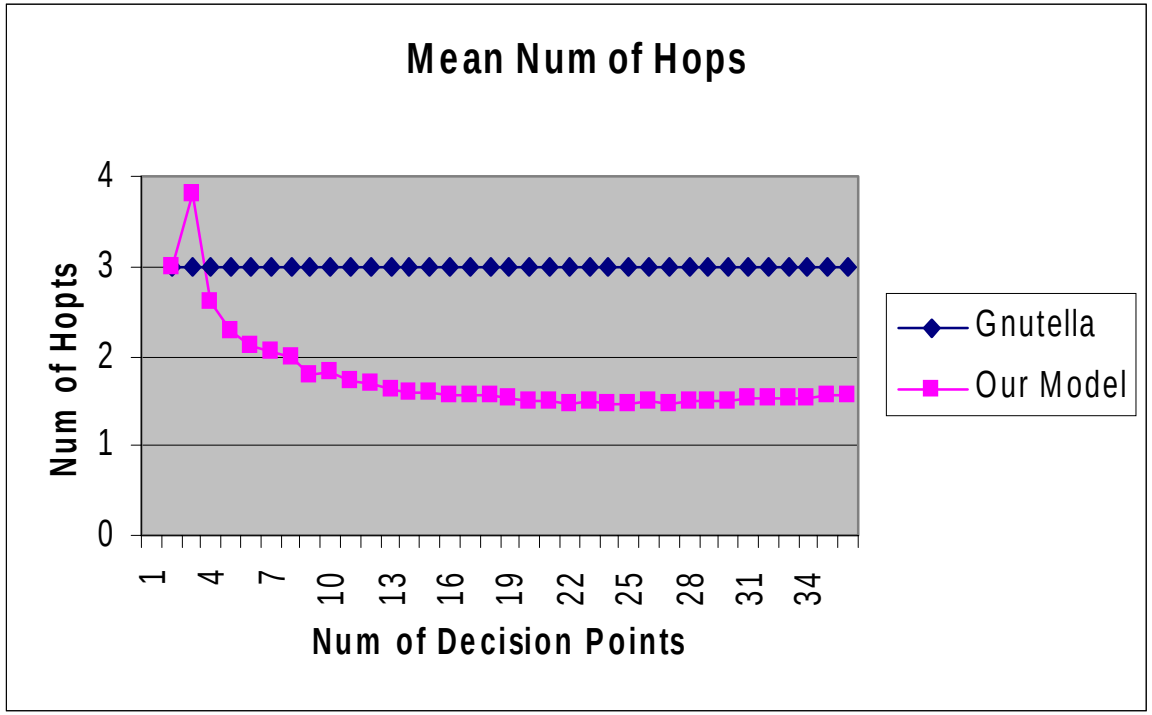


12 peers
3 connections/peer
horizon = 3

Measured num of replies to user queries

Number of replies (QueryHit msgs) grows dramatically

Performance Results (cont.)



Measured num of hops messages traverse in the network

Number of hops to find results improves

Impact of Applications at the Edges

- Objects maintained by peers not unique
- Popular objects highly replicated
- Each object characterized by meta-data
  (e.g., name, provider, keywords)
  - keyword searches
- Problem: how to uniquely identify the objects among the peers

Load Balancing

- Distribute objects on peers to meet end-to-end real-time response requirements
- Improve performance and availability of applications
- Best location to deploy objects based on
  - Number of object replicas in the network
  - Load on the peers
  - Latencies of the applications
  - Number and frequency of object invocations made by users
- Peer-to-Peer applications both originated and delivered from the edges of the network

Conclusions

- Decentralized management of peer-to-peer applications
- Connections between the peers established dynamically

Future Work

- Global state of system cannot be captured
- Current work has focused in the sharing of “small” objects
- Assure reliability of applications despite the unreliability of the peers
- Security issues (eg., user authentication, encryption..)